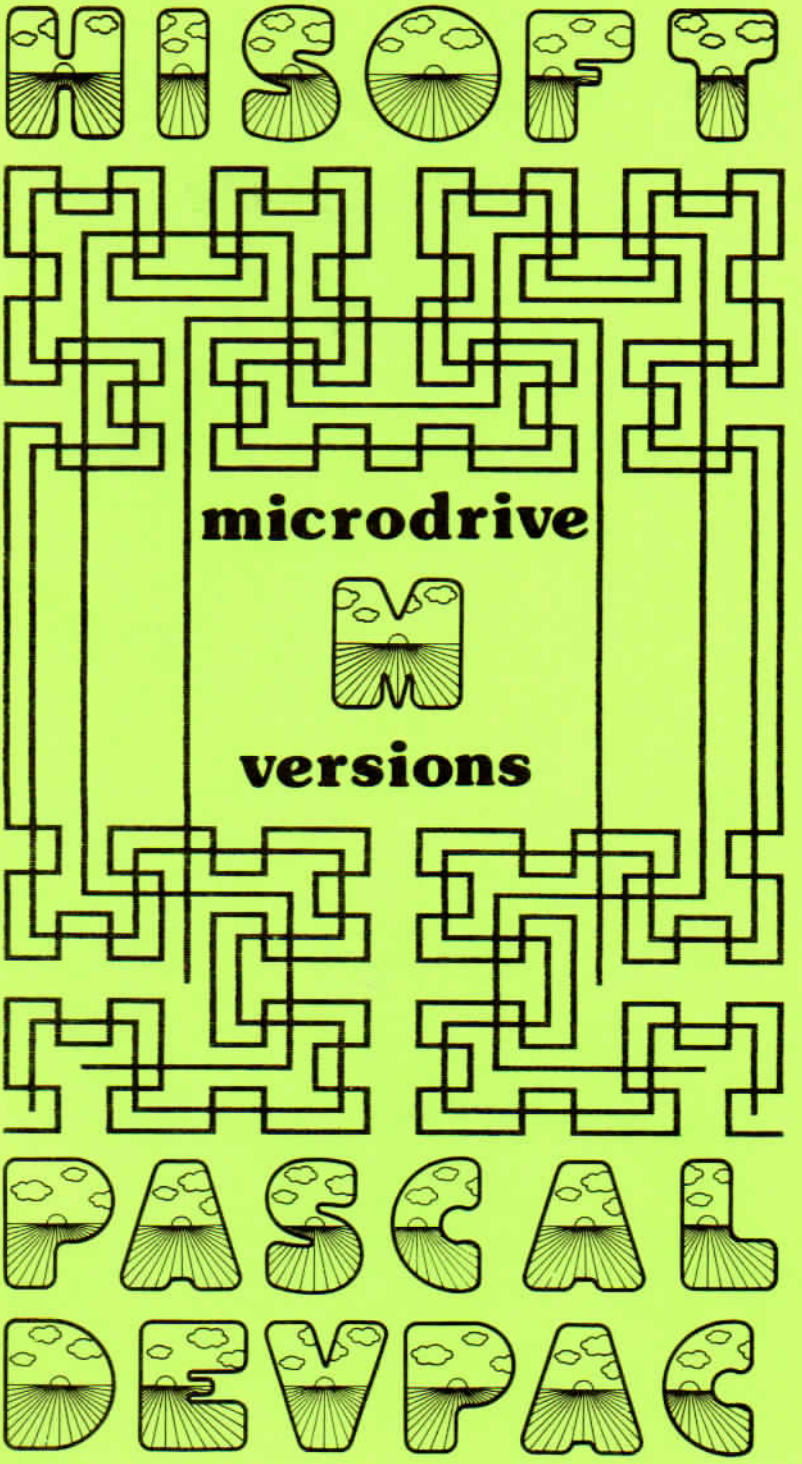
HISOFT
microdrive
M
versions
PASCAL
DEVPAC

HISOFT
13 Gooseacre Cheddington
Leighton Buzzard Beds.
LU7 0SR
Tel: (0296) 668995

HISOFT PASCAL 4T - ZX SPECTRUM VERSION

Unfortunately, Version 1.5M of Hisoft Pascal 4T (Spectrum) supplied between 14 November 1983 and 21 November 1983 inclusive, contains some errors viz:

1. Printer support via CHR(16) or $P is inoperative.
2. The 'W' editor command fails to function correctly.
3. Writing of control codes to change the temporary video attributes does not function correctly.
4. The 'G' editor command will only load the first file on a tape.

Errors 1,3 and 4 have been corrected on tapes available from 22 November 1983. Any owners of faulty copies of HP4T may return their tape and obtain a corrected tape FREE OF CHARGE. A faulty version may be detected by the message 'Bytes: HP4T5M' while loading, corrected versions display 'Bytes: HP4T15M'.

The remaining bug, listed 2 above, may be corrected in the following way:

a. Load up the HP4T15M (corrected version) from tape and answer the 'Top of RAM?' etc. messages in the normal way.

b. Enter the following Pascal program using I10,10:

```
10 PROGRAM A;
20 VAR  I,J : INTEGER;
30 BEGIN
40  I := PEEK(#8148,INTEGER);
50  J := PEEK(#814B,INTEGER);
60  POKE(#8148,J);
70  POKE(#814B,I)
80 END.
```

c. Compile and run the program as directed in the HP4T Programmer's manual.

d. Now save the corrected version of the Pascal to tape using:

```
SAVE "HP4T15M" CODE 24598,19561
```

You should note that the length of a configured version of HP4T15M (from 22 November) is 19561 , the address for a cold start entry point is 24598 and the address for a warm start entry point is 24603.

Hisoft apologises for the errors in Hisoft Pascal - we hope that you will enjoy using the package.

Hisoft is pleased to announce ZX Microdrive-compatible versions of both Pascal and Devpac - version numbers 1.5M and 3M respectively.

These versions are available from the 14 November 1983 and give the user the ability to store and retrieve textfiles to and from the Microdrives, 'include' text from a Microdrive cartridge whilst compiling or assembling and, for Pascal only, save the resultant object code onto a Microdrive cartridge. The prices for upgrading from earlier releases of Pascal and Devpac are £3 plus VAT in both cases.

We are working on even more powerful versions of these programs which will include many extra Microdrive-related features such as sequential FILE handling (for Pascal) and easy assembly from Microdrive to Microdrive in one step. These versions will be available early in the New Year of 1984.

The remainder of this document details the Microdrive extensions to Hisoft Pascal 4T Version 1.5 and Hisoft Devpac Version 3 that are currently (from 14 November 1983) available.

© Copyright Hisoft 1983.

All Rights Reserved Worldwide. No part of this publication may be reproduced or transmitted in any form or by any means, including photocopying and recording, without the written permission of the Copyright holder. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature.

It is an infringement of the Copyright pertaining to Hisoft Pascal 4T and Hisoft Devpac 3 and associated documentation to copy, by any means whatsoever, any part of Pascal 4T or Devpac 3 for any reason other than for the purpose of making one security back-up copy of the object code.

HISOFT DEVPAC VERSION 3M

Loading and Making Back-up Copy.

The loading procedure for both GENS3M and MONS3M is identical to that given in the Devpac 3 Programmer's manual except that you should allow a little more memory space for both packages. Once you have loaded either GENS3M or MONS3M (without executing them) you can make a back-up copy to either cassette tape or Microdrive cartridge as follows:

GENS3M

SAVE "GENS3M" CODE XXXXX,9046 to cassette

SAVE *"M";1;"GENS3M" CODE XXXXX,9046 to Microdrive

MONS3M

SAVE "MONS3M" CODE XXXXX,5800 to cassette

SAVE *"M";1;"MONS3M" XXXXX,5800 to Microdrive

where: XXXXX is the address at which you loaded the program.

Added Features

There are no added Microdrive-related features within MONS3M, the code has simply been altered so that it is possible to single-step code that uses the Interface 1 ROM.

The extra features available in GENS3M (compared with GENS3) are as follows:

The Editor 'P' and 'G' commands.

These commands remain identical for 'P'utting or 'G'etting text to and from cassette tape but have been modified so that, if the first 2 characters of the filename consist of a digit (1-8 inclusive) followed by a colon ':', then the text will be stored to or retrieved from the relevant Microdrive. Thus, to store lines 10 to 190 of a textfile to Microdrive 1 you simply use:

P1,190,1:TEST

and the textfile will be stored on Microdrive 1 with the name 'TEST'.

Note that, when 'P'utting, if the file already exists then it is automatically deleted whereas if, when 'G'etting, the file does not exist then the message 'File Absent' is displayed.

You must specify a filename if you wish to 'G'et a file from a Microdrive cartridge - you cannot load the first file since there isn't one!

If you do not have Microdrives attached to your system then you must ensure that the second character of any filename you use is not a colon.

The '*F' Assembler Command.

'*F' is used to 'include' text from a storage device while assembling. In GENS3M this command has been extended to allow you to include text from a Microdrive cartridge - again the method is simply to include a Microdrive number (1-8 inclusive) before the filename. On cassette you must use the 'W' editor command to dump out text that you wish to 'include' at a later stage and the use of '*F' automatically turns off all other assembler commands. On Microdrive these two restrictions can be relaxed - you 'include' a textfile that was stored on cartridge using the 'P' command and all assembler commands are available while including. Thus, to include the textfile 'TEST' (see the example above) you might have something like this:

```
                LD    A,B
                CALL  AOUT

*H Include the file 'TEST' from Microdrive

                *F 1:TEST

*H Next Routine
```

Note that you must specify a filename when using '*F' with Microdrives and, if the included code contains an error, you must continue with the assembly - do not try to get back to the editor (via 'E') while including text from a Microdrive.

The Editor 'V' command.

This command has been modified to display the current delimiter (a comma initially) as well as the default line numbers and strings. It is useful to know what the delimiter is if you have inadvertantly changed it using the 'S' command.

HISOFT PASCAL 4T VERSION 1.5M

Loading and Making a Back-up Copy.

To load Pascal 1.5M into the Spectrum simply use 'LOAD ""' (LOAD is on the 'J' key) and start your tape recorder. Once loaded the program will run automatically and you will be prompted with 'Top of RAM?' - refer to Section 0 of the Programmer's Manual for details of how to answer this and the two following questions. Normally, pressing ENTER in response to the prompts will suffice.

Once you are in the editor mode (there will be a '>' prompt on the left hand side of the screen) you can make a back-up copy of the Pascal by using the 'B'ye command to return to BASIC and then either:

SAVE "HP4T15M" CODE 24598,19570 for cassette tape

or

SAVE *"M";1;"HP4T15M" CODE 24598,19570 for Microdrive

This will save a configured version of the Pascal which you can load back into the machine simply using LOAD "" CODE or LOAD *"M";1;"" CODE. Once you have re-loaded it in this way then you must enter HP4T15M via either RANDOMIZE USR 24598 (for a cold start, destroying any text) or RANDOMIZE USR 24603 (for a warm start which preserves any existing text. You may re-enter the Pascal from BASIC using these addresses at any time.

You should note that HP4T Version 1.5M does not have a BASIC loader and thus any reference to this loader in your Implementation Note should be disregarded. Specifically, to re-enter HP4T at any time, you do not GOTO 9 or GOTO 12 as in the Implementation Note; instead you use the cold and warm start addresses given above. In addition you should ignore the instructions on making a back-up copy given in the Implementation Note, use those above instead.

Also, because there is no BASIC interface program, the cassette commands 'P' and 'G' take effect immediately and display 'Busy..' as stated in the HP4T Programmer's Manual.

We now look at the extra features that have been implemented in HP4T Version 1.5M.

The Editor 'V' Command.

This command has been modified to display the current delimiter (initially a comma ',') as well as the default line numbers and strings.

The Editor 'P' and 'G' commands.

These commands remain identical for 'P'utting or 'G'etting text to and from cassette tape but have been modified so that, if the first 2 characters of the filename consist of a digit (1-8 inclusive) followed by a colon ':', then the text will be stored to or retrieved from the relevant Microdrive. Thus, to store lines 10 to 190 of a textfile to Microdrive 1 you simply use:

P1,190,1:TEST

and the textfile will be stored on Microdrive 1 with the name 'TEST'.

Note that, when 'P'utting, if the file already exists then it is automatically deleted whereas if, when 'G'etting, the file does not exist then the message 'File Absent' is displayed.

You must specify a filename if you wish to 'G'et a file from a Microdrive cartridge - you cannot load the first file since there isn't one!

If you do not have Microdrives attached to your system then you must ensure that the second character of any filename you use is not a colon.

The '$F' Compiler Option.

While compiling, you may 'include' a textfile from either cassette tape or from a Microdrive by using the '$F' compiler option.

The format of this option when used to compile a file from cassette tape is explained in Section 3 of the HP4T Programmer's Manual. When used with Microdrives, the only difference is that the first two characters of the filename should be a Microdrive number (1-8 inclusive) followed by a colon ':'.

Thus to include the file 'TEST', stored to Microdrive in the example above, you could use the following:

```
10 PROGRAM A;
20 BEGIN
30
40 {$F 1:TEST    }
50
60 END.
```

Note that, to include a file from cassette tape, you must have used the 'W' editor command to dump out the text in a blocked format whereas, with Microdrives, you include a textfile saved using the normal 'P'ut command.

Do not try to return to the editor (via 'E' or 'P') if there is a compilation error in a file that you are including - you must continue the compilation.

TIN and TOUT.

Data structures may be saved to cassette or Microdrive using the TIN procedure from within a Pascal program and retrieved using TOUT - see Sections 2.3.5.8 and 2.3.5.9 of the HP4T Programmer's Manual for details. To direct data to or from Microdrives simply start the required filename with a Microdrive number (1-8 inclusive) and a colon ':' e.g.

```
10  PROGRAM A;
20  VAR  A,B : ARRAY[1..20] OF CHAR;
30  BEGIN
40  {some code to set up the array A}
50
60   TOUT('1:ARRA   ',ADDR(A),SIZE(A));
70
80  {some more program}
90
100  TIN('1:ARRA   ',ADDR(B));
```

```
110 (etc. etc. )
```

'T'ranslation of Object Code.

The editor 'T' command can be used to 'translate' the Pascal object code so that it may be stored on cassette tape or Microdrive. Remember that, when translating, the compiler is deleted and the runtimes together with the object code are saved to the storage device. Thus you must reload the compiler after a translation if you wish to compile some more text - however this is not the usual practice since you will only save complete and tested object code.

To translate the object code to cassette use the command as specified in Section 4.2.5 of the HP4T Programmer's Manual. To reload the object code from tape simply use LOAD "" CODE and then, to execute it, use RANDOMIZE USR 24608.

To translate the object code to Microdrive use the same 'T' command as in Section 4.2.5 but put a Microdrive number and a colon at the front of the filename e.g. T1,9999,1:OBJ . The runtimes and object will then be stored on the relevant Microdrive as a PRINT file. To reload this file you must have some sort of loader. We give below a typical PRINT file loader which you can enter with your favourite assembler (DEVPAC we hope!) or hand code - this loader prompts you for the name of the file, loads it off drive 1 and executes 24608. You can modify it to load from different drive numbers, perform better error checking etc. We will supply this loader on cassette with the next version of HP4T.

```
*H An HP4T Print File Loader for Microdrives

        EXX
        PUSH HL
        EXX
        LD   IY,#5C3A
        LD   A,2          ;open video
        CALL #1601
        LD   HL,MSGN
        LD   B,NAMEL
NLOOP   LD   A,(HL)       ;put out prompt
        INC  HL
        RST  16
        DJNZ NLOOP
        LD   A,1          ;open keyboard
        CALL #1601
        LD   HL,NAME      ;get name from keyboard
        LD   C,0
KLOOP   LD   DE,NAME
KBD     CALL #15E6
        JR   Z,KBD
        JR   NC,KBD
        CP   13           ;end of name?
        JR   Z,NEND
        CP   12           ;backspace?
        JR   Z,NBACK
        LD   (HL),A
        RST  16           ;echo character
        INC  HL
        INC  C
        JR   KLOOP
NBACK   SBC  HL,DE
        ADD  HL,DE
```

```
        JR    Z,KLOOP
        RST   16
        DEC   HL
        DEC   C
        JR    KLOOP
NEND    PUSH  BC
        RST   8             ;create new variables
        DEFB  #31
        POP   BC
        EI
        LD    HL,NAME       ;initialise workspace
        LD    (#5CDC),HL
        LD    H,0
        LD    L,C
        LD    (#5CDA),HL
        LD    HL,1          ;drive number
        LD    ( #5CD6),HL
        RST   8             ;open the file
        DEFB  #22
        XOR   A             ;initialise CHREC
        LD    (IX+13),A
        RST    8            ;get first record
        DEFB  #27
        LD    DE,24608      ;destination address
MORE    PUSH  DE
        LD    DE,82
        PUSH  IX
        POP   HL
        ADD   HL,DE         ;address the data
        POP   DE
        LD    C,(IX+69)     ;get length of data
        LD    B,(IX+70)
        LDIR                ;move data
        BIT   1,(IX+67)     ;end of file?
        JR    NZ,END
        PUSH  DE
        RST   8             ;get next record
        DEFB  #25
        POP   DE
        JR    MORE
END     RST   8             ;close file
        DEFB  #23
        EXX
        POP   HL
        EXX
        JP    24608         ;execute code

MSGN    DEFM  "Name: "
NAMEL   EQU   $-MSGN

NAME    DEFS  10            ;assume maximum 10 characters
```

Alternatively, instead of using the above loader, you can note down the end address of the object code from the compiler listing and, remembering that it starts at 24608, SAVE the object code to Microdrive using BASIC's SAVE "" CODE construct. You may well find this a more convenient solution although the loader given above is bound to come in useful!

HISOFT PASCAL 4TM IMPLEMENTATION NOTE
48K ZX SPECTRUM

Loading HP4TM From Tape.

Unpack the cassette tape from its case and load it into your cassette recorder with Side A (the side with the label) uppermost. On your SPECTRUM make sure that you are in Keyword Entry Mode and then enter:

LOAD "" (press J and then " twice)

Now press PLAY on the tape recorder: first a small BASIC loader will be loaded, this will execute automatically and proceed to load the HP4TM code. If a tape error is detected then stop the tape, rewind to the start, press NEW (on the 'A' key) on the SPECTRUM and then enter LOAD "" again. If you still get a tape loading error then try adjusting the volume on your tape recorder; if errors persist please return the tape to Hisoft and we will replace it.

Once the HP4TM code has been loaded it will execute automatically and the message 'Top of RAM?' will be displayed - now consult Section 0.0 of the Hisoft Pascal 4T Programmer's Manual for details of how to proceed.

Implementation on the SPECTRUM.

The ZX SPECTRUM is a rather unusual computer and, to a certain extent, the implementation of HP4T reflects this. The various control codes discussed in the Programmer's Manual are reached as follows on the SPECTRUM:

RETURN	via the 'ENTER' key.
CC	via CAPS SHIFT and 1.
CH	DELETE i.e. CAPS SHIFT and 0.
CI	via CAPS SHIFT and 8.
CP	via CAPS SHIFT and 3 enabling 'L'isting of text to printer.
CX	via CAPS SHIFT and 5.
CS	via CAPS SHIFT and SPACE.

The ZX SPECTRUM keyword entry scheme is not supported (we see this as a positive advantage), instead all text must be inserted using the normal alphanumeric keys. Using SYMBOL SHIFT and any key (except I) will always reach the ASCII symbol associated with that key and not the keyword e.g. SYMBOL SHIFT T gives '>' and SYMBOL SHIFT G gives '}'. You must not use the single symbols <=, <> and >= ; instead these should be entered as a combination of the symbols <, > and =.

The editor comes up in upper case mode, this may be toggled in the normal way using CAPS SHIFT and 2.

You have control over the temporary attributes of the various character positions on the screen through the use of the standard control codes (e.g. WRITE(CHR(17),CHR(4)) will make the 'paper' green) but you cannot change the permanent attributes. If, while using these control codes, an invalid sequence is detected then the message 'System Call error' will be displayed and the execution aborted. Note that certain CHR codes are interpreted by HP4T (e.g. CHR(8) is taken as DELETE) and thus these codes cannot be passed directly to the Spectrum ROM - use the SPOUT procedure (page 61) if you want to write CHR codes without interpretation by HP4T.

When dumping text or object code to tape you must be careful to have the tape recorder in RECORD mode before beginning the dump.

If you have used the 'T'ranslate command to save the object code and runtimes on tape then to load the program simply enter LOAD "" CODE from within BASIC. To execute the program enter RANDOMIZE USR 24608 from within BASIC.

From within ZX BASIC, you can re-enter the HP4TM editor in one of two ways: enter RANDOMIZE USR 24603 to perform a warm start i.e. preserving the Pascal program or RANDOMIZE USR 24598 to do a cold start, re-initialising the Pascal and clearing any existing Pascal text.

The ZX Printer is supported via the use of the compiler 'P' option (see Section 3.2 of the Programmer's Manual) and via CHR(16) in a WRITE or WRITELN statement. Note that, as a result, you cannot use CHR(16) within a WRITE(LN) statement to specify the INK colour - instead you can use CHR(15) to set the INK.

To make a back-up copy of HP4TM proceed as follows:

1. Load HP4TM from tape and answer the 'Top OF RAM?' etc. messages normally.

2. Return to BASIC using the editor's B command.

3. Use SAVE "HP4T15M" CODE 24598,19558 to save the Pascal to tape.

4. You can subsequently use LOAD "" CODE to reload the compiler into the Spectrum but note that you must then enter it only via RANDOMIZE USR 24598 (for a cold start) or RANDOMIZE USR 24603 (for a warm start).

Note that you are authorised by Hisoft to make only one working copy.

Please do not hesitate to contact us if you experience any difficulty with Hisoft Pascal 4 - we can only solve the problems if we know what they are!